

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**Desarrollo de Chatbot
para el Apoyo a cursos de
Programación de Primer Año
“Probot”**

Bastian Tobar Mori

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Diciembre 2018

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**Desarrollo de Chatbot para el Apoyo a cursos de Programación de Primer
Año
“Probot”**

Bastian Tobar Mori

Profesor Guía: **Ismael Figueroa Palet**

Profesor Co-referente: **Laura Griffiths Molina**

Carrera: **Ingeniería de Ejecución en Informática**

Diciembre 2018

Agradecimiento a mis padres William y Paula quienes fueron pilar fundamental en todo este camino, sin su ayuda y su apoyo en todo momento esto no hubiera sido posible. A mis hermanas Antonia y Sofía por ser las buenas hermanas que han sido siempre conmigo acompañarme cuando las necesito. A mis abuelos Oscar y Palmira por darme su cariño y festejar mis triunfos como si fueran de ellos.

A mis amigos que siempre estuvieron desde que comenzó esta aventura ya sea con su buena onda, con su ayuda académica, sus consejos en otras cosas. Claudio, Elías, Nicolás, Cristian, Esteban, Marcelo y aun gran amigo de la vida que ya no está entre nosotros, pero sigue siempre en mi memoria Rodrigo que siempre confió en mí en que lo podría lograr.

A mi profesor Osvaldo Rodríguez por sacar mi mejor versión y enseñarme a dar lo mejor de mí en todo momento y jamás bajar los brazos sin importar la situación.

Finalmente, gracias a mi pareja Yocelin por llegar a mi vida a apoyarme en todo momento y darme ese impulso a seguir y ser una mejor persona, gracias por ser parte de este proceso.

Todo esto fue posible gracias a ustedes.

Índice

Resumen.....	iii
Abstract	iii
Lista de Figuras	iv
Lista de Tablas	v
1 Introducción	1
2 Marco general de Probot	2
2.1 Objetivos.....	2
2.1.1 Objetivo general.....	2
2.1.2 Objetivos específicos.....	2
3 Situación de estudio	3
3.1 Descripción de la situación de estudio	3
3.2 Estado del arte	3
3.2.1 Dialogflow	4
3.2.2 Converse.ai.....	6
3.2.3 Pandorabots	7
4 Problemas Comunes en el Desarrollo de Chatbots	8
4.1 Conflictos	9
4.2 Ambigüedad	9
4.3 Errores gramaticales	9
4.4 Ortografía.....	9
4.5 Uso de anáforas.....	10
4.6 Lenguaje coloquial	10
4.7 Anglicismo	11
4.8 Uso de chatbot en educación	11
5 Propuesta de solución	11
5.1 Plan de trabajo tentativo:	12
5.2 Encuesta mixta.....	12
5.3 Herramientas de desarrollo	12
5.3.1 Dialogflow	12
5.3.2 Pandorabots	13
5.4 Marcha blanca de Probot	13
6 Diseño.....	13
6.1 Interfaz	13
6.1.1 Angular	13
6.1.2 Bootstrap.....	14
6.1.3 Font Awesome	14
6.2 Base de datos	14

6.2.1 Firebase.....	14
6.3 Pantallazos Probot	14
7 Desarrollo del software.....	15
7.1 Obtención de base de conocimientos	15
7.2 Implementación de objetivo general.....	15
7.3 Implementación de objetivos específicos.....	15
8 Alcances y limitaciones	17
8.1 Alcances	17
8.2 Limitaciones	18
8.3 Trabajo a futuro	18
9 Conclusiones	18
10 Bibliografía:	19
Anexos.....	21

Resumen

Este documento, muestra uno de los problemas más recurrentes cuando el docente enseña nuevos conceptos y/o conocimientos a los alumnos de la escuela de ingeniería de informática de la PUCV en un ramo de primer año, fundamentos de programación. El problema detectado y al cual se pretende dar solución, es que los alumnos pueden quedar con dudas y/o inquietudes que no son atendidas a tiempo.

Para esto se propone desarrollar un chatbot (Probot) donde, debemos destacar que la principal función es poder aclarar estas dudas conceptuales, que le puedan surgir al estudiante y que pueda ser un apoyo para los docentes y así realizar apoyar la enseñanza.

Palabra clave: Probot.

Abstract

the teacher teaches new concepts and/or knowledge to the students of the computer engineering school of the PUCV in a first-year field, fundamentals of programming. The problem detected and which is intended to solve is that students can be left with questions and/or concerns that are not addressed in time.

For this purpose, it is proposed to develop a chatbot (Probot) where we must emphasize that the main function is to clarify these conceptual doubts, that may arise to the student and that can be a support for teachers and thus make a complete teaching.

Keyword: Probot.

Lista de Figuras

Figura 3.1 Ejemplo de cómo Dialogflow se comporta ante la entrada de un usuario	4
Figura Anexo Pantallazo pantalla inicial Probot.....	30
Figura Anexo Pantallazo pantalla introductoria Probot.....	30
Figura Anexo Pantallazo pantalla feedback Probot.....	30

Lista de Tablas

Tabla 7.3.1 Implementación objetivos específicos.....	17
--	----

1 Introducción

En el presente documento se presenta el estado del ramo Proyecto de Título (INF 4541-7), donde se verá a lo largo de este informe, la investigación realizada, la cual conlleva la situación de estudio, el problema detectado, estado del arte y una propuesta de la solución ante este problema detectado. En términos generales, el punto central del proyecto es el desarrollo de un chatbot (**Probot**), por lo que se presentan el marco conceptual y las herramientas que se emplearán para llevar a cabo su desarrollo y su posterior ejecución.

En términos más concretos, Probot apunta a resolver una problemática detectada en los cursos de programación de primer año de la Escuela de Ingeniería Informática de la PUCV, particularmente en la asignatura Fundamentos de Programación. Esta solución consiste en el desarrollo de un asistente virtual (en este caso chatbot), que cada vez está más presente en nuestro diario vivir y con el paso del tiempo su presencia irá en aumento gracias al impulso que actualmente están aportando las nuevas tecnologías que se desarrollan día a día. Pero ¿qué son los chatbots y que es lo que hacen? haciendo una breve descripción podemos decir que los chatbots son asistentes virtuales que son utilizados principalmente para llevar a cabo las funciones de atención al cliente, por lo que pueden resolver las necesidades de una persona.

2 Marco general de Probot

A continuación, se presenta el contexto en el que se desarrolla ProBot, describiendo la situación general del mismo, así como el objetivo general, y los objetivos específicos a realizar.

2.1 Objetivos

En esta sección se presenta una descripción del objetivo principal del proyecto y los objetivos específicos que se deben llevar a cabo para el cumplimiento de éste.

2.1.1 Objetivo general

Desarrollar un chatbot (Probot), utilizando frameworks y librerías existentes, que soporte consultas básicas de estudiantes que se encuentren en el curso de programación de primer año para servir como herramienta de apoyo.

2.1.2 Objetivos específicos

- Identificar principales dudas o inquietudes de los estudiantes de primer año sobre programación en el lenguaje C, y determinar su aplicabilidad como parte del Chatbot.
- Determinar las preguntas que serán respondidas con el modelo de Entidades, y las que serán respondidas por el modelo de Flujo Conversacional.
- Diseñar las entidades y flujos conversacionales según corresponda a cada pregunta.
- Elaborar Prototipo de chatbot que resuelva principales dudas respecto a programación.
- Evaluar de manera experimental la apreciación de los estudiantes en cuanto a la utilidad del Chatbot, mediante su aplicación en actividades controladas.

3 Situación de estudio

En la siguiente sección se presenta una pequeña descripción de la situación actual en cuanto al uso de chatbot y el problema que se pretende dar solución con su uso. Además, algunos problemas que podrían surgir en el uso del chatbot.

3.1 Descripción de la situación de estudio

Frecuentemente dentro del mundo académico, cada vez que un estudiante se enfrenta a una cátedra de diversa área, al finalizar ésta puede quedar con una o muchas dudas, los que puede llevar como resultado a un mal rendimiento académico, ya que en la curva de aprendizaje del cualquier concepto y/o materia surgen dudas, inquietudes entre otras incógnitas que no son aclaradas en el momento, ya sea por falta de tiempo del docente a cargo o por miedo del estudiante a preguntar, entre otros. Dicho lo anterior esto afecta directamente al rendimiento del estudiante en sus calificaciones.

Dado que este problema es habitual en el aprendizaje del estudiante, el objetivo de este proyecto es desarrollar un prototipo de Probot como herramienta de apoyo para los estudiantes, de manera de poder contestar estas dudas, cuando el estudiante lo requiera sin importar la hora y el lugar. Se espera también analizar las consultas realizadas al Probot como retroalimentación a los docentes sobre puntos a reforzar en la asignatura.

3.2 Estado del arte

Los chatbots son agentes conversacionales que interactúan con usuarios utilizando lenguaje natural. Estos están disponibles para ser utilizados en diferentes dominios, como marketing, psicología, informática entre otros. Sin embargo, su base de conocimiento está codificado a mano en su “cerebro” [1]. Primeramente, los chatbot se basaban en el uso de reglas diseñadas a mano, pero en la década de los noventa, con el desarrollo del procesamiento del lenguaje natural (NLP por su sigla en inglés para *Natural Language Processing*), los chatbot evolucionaron, introduciendo algoritmos de aprendizaje automático para el procesamiento del lenguaje natural. El esquema de NLP también posee distintos niveles de procesamiento, entre los cuales tenemos la morfología, fonología, lexicología, sintáctica y semántica.

Los chatbots son una categoría de los agentes conversacionales, los cuales son sistemas de software que imitan interacciones con gente real. Los agentes conversacionales, por su parte, son una clase de sistemas de diálogo que han sido sujetos a investigación en comunicación por décadas. Existen muchas plataformas para la construcción de chatbots sin tener que escribir código, o sin tener que ser necesariamente un investigador experto en NLP, tales como: Dialogflow, Converse.ai, PandoraBot, Rasa.ai, Chatterbot entre otras, estas permiten a los diseñadores crear flujos conversacionales utilizando elementos visuales. A continuación, se describen algunas plataformas.

3.2.1 Dialogflow

Dialogflow[2] permite construir interfaces conversacionales sobre sus productos y servicios al proporcionar un poderoso motor de comprensión del lenguaje natural (NLU Natural language understanding) para procesar y comprender la entrada de lenguaje natural. Las interfaces informáticas tradicionales requieren una entrada estructurada y predecible para funcionar correctamente, lo que hace que el uso de estas interfaces sea poco natural y, a veces, difícil. Si los usuarios no pueden entender fácilmente esta entrada estructurada, les será difícil descifrar qué hacer para dialogar con el chatbot.

Para entender más el comportamiento de Dialogflow considere la siguiente solicitud de un usuario: "¿Cuál es el pronóstico de hoy?". Otros usuarios también pueden preguntar:

- "¿Cómo está el clima ahora?"
- "¿Cómo es la temperatura en San Francisco mañana?"
- "¿Cómo será el clima el día 21?"

Ejemplo tomado de la documentación de Dialogflow [3].

Incluso ante esta simple pregunta, se puede ver que las experiencias conversacionales son bastante complejas de poder implementar. La interpretación y el procesamiento del lenguaje natural requiere un analizador de lenguaje muy sólido el cual sea capaz de comprender los diferentes matices del lenguaje. En este sentido, Dialogflow nos da la facilidad de lograr fácilmente una experiencia de usuario conversacional manejando la comprensión del lenguaje natural (NLU) para liberar al desarrollador de esta implementación. Dialogflow funciona en base a *agentes* que pueden comprender los vastos y variados matices del lenguaje humano y traducirlo al significado estándar y estructurado que sus aplicaciones y servicios pueden comprender. Ahora veamos cómo Dialogflow podría manejar los ejemplos anteriores para las solicitudes de pronóstico del tiempo.

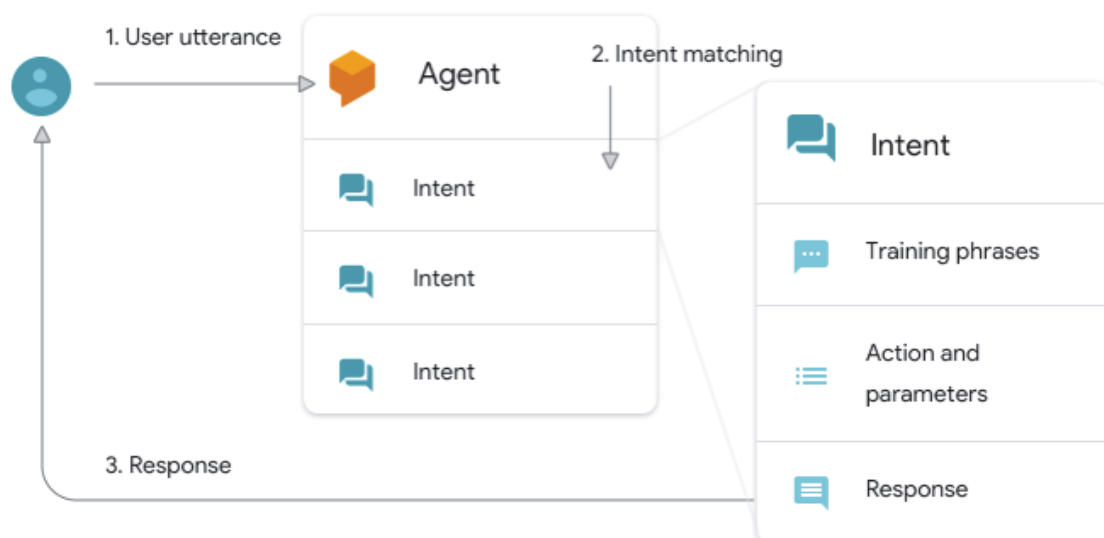


Figura 3.1 Ejemplo de cómo Dialogflow se comporta ante la entrada de un usuario [3].

Para buscar un pronóstico del tiempo, es posible que necesite algunas piezas de información, como el tiempo que los usuarios quieren para el pronóstico y su ubicación. Sin embargo, como mencionamos anteriormente, diferentes usuarios pueden solicitar un pronóstico de diferentes maneras. Dialogflow puede comprender estas diferencias y traducirlas a la intención del usuario estándar de obtener el pronóstico. A continuación, puede analizar la solicitud del usuario de los datos pertinentes que necesita para completar la solicitud. En este caso, ese es el tiempo y la ubicación deseados por el usuario para el pronóstico del tiempo. En términos generales Dialogflow principalmente está constituido por un agente, intención y entidades, ya manejando esto de manera básica se puede construir un chatbot básico.

3.2.1.1 Agente

Ayuda a procesar la entrada del usuario en datos estructurados que puede usar para devolver una respuesta adecuada. El desarrollador define todos estos componentes dentro de uno o más **intentos o intención** (del inglés *intent*), que definen cómo asignar la entrada del usuario a una respuesta correspondiente.

Una intención básica se compone de:

- **Frases de entrenamiento:** define frases de ejemplo de lo que los usuarios pueden decir. Dialogflow utiliza estas frases de capacitación y, naturalmente, las expande a muchas más frases similares para crear un modelo de idioma que coincida exactamente con la información del usuario. A través de la capacitación adicional y el aprendizaje automático, Dialogflow crea un modelo de lenguaje más sólido y variado para que coincida mejor con la información del usuario. ¿Por ejemplo?

- **Acción y parámetros:** para mejorar el modelo de lenguaje de un intento, también puede anotar sus frases de entrenamiento con *entidades* o categorías de datos que desea que coincida Dialogflow. Esto le permite indicar a Dialogflow que desea un tipo particular de entrada y no solo hacer coincidir la entrada literal del usuario. Dialogflow extrae entidades coincidentes como parámetros de las frases de entrenamiento. A continuación, puede procesar estos parámetros en lógica llamada *cumplimiento* para personalizar aún más una respuesta al usuario. Aprenderá sobre el cumplimiento más adelante en este documento.

- **Respuestas:** define una respuesta de texto, voz o visual para el usuario, que generalmente solicita a los usuarios de una manera que les permita saber qué decir a continuación o que la conversación está por terminar. Para enviar respuestas, puede usar el controlador de respuesta incorporado de Dialogflow o el cumplimiento de llamadas para procesar los datos extraídos y devolver una respuesta a Dialogflow.

3.2.1.2 Entidades

Son el mecanismo que utiliza Dialogflow para identificar y extraer datos útiles de las entradas de lenguaje natural. Si bien los intentos le permiten a su agente comprender la motivación detrás de la entrada de un usuario en particular, las entidades se utilizan para seleccionar piezas específicas de información que los usuarios mencionan, desde direcciones postales a nombres

de productos o montos con unidades. Cualquier información importante que desee obtener de la solicitud de un usuario tendrá una entidad correspondiente.

3.2.2 Converse.ai

Converse.ai [4] es una herramienta que posee un NLP *Parsing Engine* (Análisis de los elementos del motor) incorporado y tiene la capacidad de crear plantillas de conversación. Converse.ai no es una plataforma de mensajería, sino que es una plataforma de automatización que se adapta a los servicios de mensajería existentes (canales). La herramienta está diseñada para manejar una amplia gama de casos de uso tales como, consultas previas a la venta, servicio al cliente, comercio conversacional, compromiso post venta, consultas previas a una venta, re-compromiso y evolución del embudo de ventas. Para entender de mejor forma el funcionamiento de Converse.ai necesitaremos una serie de elementos para completar una conversación. Estos elementos serán útiles para contextualizar y dar coherencia a lo recibido por el chatbot, con el fin de poder abordar una respuesta con los resultados esperados. Para la solución de esto, será necesario un manejo de respuestas que permita gestionar una conversación a través de plantillas, las cuales se formulan por frases.

3.2.2.1 Plantillas

Dentro de Converse.ai una plantilla de conversación es una pieza clave de funcionalidad, la que nos indica qué hacer cuando una frase coincide con ella. Ésta puede configurarse para ofrecer una respuesta o para iniciar una conversación con un usuario. Las plantillas de conversación se dividen en estados o pasos. Para cada paso, Converse puede hacer una pregunta o entregar una respuesta, al finalizar estos pasos la conversación habrá finalizado y si está configurada desencadenará un **WebHook**, que corresponde a un método de alteración del funcionamiento de una página o aplicación web, con **callbacks** (devolución de llamadas) personalizados.

Converse.ai posee una serie de plantillas las cuales se activan mediante el ingreso de texto del usuario, es decir un evento. Para activar las plantillas hay que entrenar al bot mediante una serie de frases, el uso de sus capacidades de lenguaje natural puede coincidir con alguna variedad. Respecto a las plantillas avanzadas podemos encontrar: **Handlebars**, el cual puede utilizar contenido dinámico del usuario o conversación para dar respuestas dinámicas. Las **Junctions** son uniones que permiten a las plantillas tener múltiples rutas dependiendo de los criterios, no hay límite de cruce. En otras palabras, una plantilla de conversación es flexible, dado que se puede escribir su propio sistema desde cero.

3.2.2.2 Conversaciones

También la herramienta permite añadir conversaciones las cuales establecen respuestas a las frases de invocación, por ejemplo:

- Usuario: Hola
- BOT: Hola, ¿En qué puedo ayudarte?

Las conversaciones pueden desarrollarse por la información ingresada por el usuario, las cuales generalmente son en respuestas a las preguntas realizadas por el bot, la cual va actualizando la información faltante de la conversación, con el fin de evitar preguntas innecesarias. En el caso de que la conversación no coincida con las plantillas existentes intentará verificar si coincide con una plantilla de conversación alternativa para cambiar la conversación. Respecto al término de una conversación, no se asegura el éxito de esta.

Si el usuario completa todos los estados requeridos, la conversación se considera completa. Por otro lado si un usuario no proporciona una entrada correcta a una pregunta individual dos veces, fallará automáticamente para evitar que el usuario caiga en un bucle, además se puede activar el disparador de escalamiento humano.

3.2.3 Pandorabots

Es una compañía de inteligencia artificial que proporciona un servicio web para construir y desplegar chatbots. Pandorabots es uno de los servicios de almacenamiento de chatbot más grandes y antiguos del mundo. La plataforma implementa y soporta el desarrollo del estándar abierto de AIML 2.0, el cual es el lenguaje más popular para crear chatbots disponibles en el mercado hasta la fecha. AIML o *Artificial Intelligence Markup Language*, permite a las personas ingresar conocimiento en chatbots. Los objetos AIML se forman por unidades llamadas temas y categorías, los cuales contienen datos tanto analizados como no analizados. Esta herramienta proporciona acceso API a su plataforma de alojamiento chatbot, la cual ofrece los siguientes SDK en GitHub: Java, Ruby, Go, PHP, Python, Node.js.

3.2.3.1 Categoría

Es la unidad básica de conocimiento en AIML, la cual consiste en una pregunta de entrada y respuesta de salida, con un contexto el cual puede ser opcional. La pregunta o estímulo se llama patrón y la respuesta se llama plantilla la cual en su forma más simple consiste en un texto sin marcar. De manera más general, las etiquetas AIML transforman la respuesta en un programa que puede guardar datos, activar otros programas, dar respuestas condicionales.

3.2.3.2 Recursión

La reducción simbólica se encarga de simplificar formas gramaticales más complejas en más simples. Para la recursión AIML implementa distintos operadores los cuales se describirán en detalle a continuación:

1. **Reducción simbólica:** reduce formas gramaticales complejas a otras más simples.
2. **Divide y conquista:** Divide una entrada en dos o más subpartes y combina las respuestas para cada una.
3. **Sinónimos:** asigna diferentes formas de decir lo mismo a la misma respuesta.
4. **Correcciones ortográficas o gramaticales.**
5. **Detección de palabras clave en cualquier lugar de la entrada.**
6. **Condicionales:** ciertas formas de ramificación pueden implementarse con <srail>.
7. **Cualquier combinación de (1) - (6)**

Las plantillas por lo general se activan mediante palabras claves. Los chatbot robustos se forman en base a un proceso cíclico de aprendizaje supervisado llamado «*Targeting*».

3.2.3.3 Palabras clave

Para que las plantillas AIML sean activadas es necesaria la aparición de una palabra clave en cualquier parte de la oración recibida por el chatbot. Para ir relacionando las palabras claves es necesario el uso de relaciones de estas palabras y las categorías, con el fin de enlazarlas de modo que todas las conjeturas generen una misma respuesta

3.2.3.4 Targeting

Es un ciclo que involucra al bot, botmaster —quien gestiona el chatbot— y cliente, en el caso de que el mensaje del usuario no coincida, entonces el bot lo registra y lo entrega al botmaster para crear respuestas adecuadas en base consultas comunes[5].

Aunque AIML puede permitir que los clientes entrenen al chatbot, esto puede ser muy arriesgado dado que no hay forma de evadir a los hackers y trolls que corrompen al bot enseñándole cosas inapropiadas. Por lo tanto, se recomienda asignar estas funciones a un grupo de usuarios de confianza.

3.2.3.5 Contexto

Para definir un contexto se utiliza la palabra clave “*that*” la cual encierra un patrón que coincide con el enunciado anterior del robot. Las categorías se activan cuando el usuario dice “Sí” por ejemplo: El chatbot pregunta: “¿Te gustan las películas?”, en el caso de que el usuario responde “Sí”, la categoría coincide y el bot responde “¿Cuál es tu película favorita?”, por lo cual el diálogo continúa en la misma línea de conversación[6].

4 Problemas Comunes en el Desarrollo de Chatbots

A continuación, se presentan los problemas que se producen con más frecuencia en el desarrollo de un chatbot, estos van ligados a directamente al procesamiento del lenguaje escrito esto hace referencia a la entrada que genera el usuario del chatbot.

4.1 Conflictos

En el contexto de la construcción de chatbots, surgen varios problemas derivados de eventos que pueden afectar en la comprensión del usuario. Estos son los problemas más comunes en la interacción humano-máquina. Algunos se deben a la costumbre de los usuarios de utilizar el chat con faltas de ortografía, o con errores gramaticales. También se consideran los conflictos en el mismo lenguaje, como la ambigüedad y el uso de anáforas o la presencia de anglicismos.

El problema tiene directa relación con el proceso de la comprensión del lenguaje natural, en el análisis semántico. Más adelante se profundizará sobre este tópico.

4.2 Ambigüedad

Se define el concepto de ambigüedad como “la peculiaridad de algunos enunciados consistente en que son susceptibles a dos o más interpretaciones”[7]. Es decir, una palabra u oración puede tener más de un sentido. Así mismo, la ambigüedad puede ser estructural o léxica.

Un ejemplo es la palabra “transacción”, la cual hace a referencia principalmente a una transferencia, a un retiro de dinero, o a un pago. Si un usuario desea hacer una transacción, deberá especificar el tipo de ésta, o caerá en la ambigüedad.

El problema tiene directa relación con el proceso del NLU, en el análisis semántico.

4.3 Errores gramaticales

El chequeo gramatical ayuda a enfrentar diferentes escenarios, y en el aprendizaje del lenguaje. Los errores gramaticales en una entrada de texto se deben principalmente al incorrecto uso del lenguaje. Es decir al uso incorrecto de las personas, números, casos o género, utilización incorrecta de los verbos, un orden erróneo de las palabras, etc.[8]

Un ejemplo de error gramatical es la expresión “¿Yo rut una cuenta tarjeta obtener puedo?”, lo que está mal redactada. Allí el orden de las palabras es el incorrecto, donde debiera decir: ¿Yo puedo obtener una tarjeta cuenta rut?

La propuesta hecha por Mozgovoy, de nombre *LanguageTool*, es un lenguaje basado en XML para describir reglas incorrectas. En su forma más simple, una mala regla es simplemente una secuencia de símbolos que se ajustará en el texto [8]. *LanguageTool* hace uso de bibliotecas de terceros para dividir y etiquetar el texto de entrada. Afortunadamente, hay varias soluciones listas disponibles para este propósito, como por ejemplo, el MXPOST de Rantparkhi[9] y MXTERMINATOR[10].

4.4 Ortografía

Según la RAE (Real Academia Española), ortografía se define como un conjunto de normas que regulan la escritura de una lengua. Es por ello, que es importante hacer el chequeo de las posibles infracciones en este aspecto. Existen dos tipos de errores, las cuales son[11]:

-non-word errors: Donde la palabra mal escrita no es válida en el idioma. (Ej: Decir “*libro*” en el idioma inglés).

-real-word errors: Donde la palabra es válida en el idioma, pero es inapropiada en el contexto, por lo tanto, no da el significado pretendido. Por ejemplo, mencionar un gato cuando el contexto es la obtención de una CuentaRUT.

Estos errores pueden deberse a estos siguientes factores:

-Error tipográfico: Es el resultado de un desliz de coordinación motriz provocando un error de tipeo. Por ejemplo, escribir tarjeat, en vez de tarjeta.

-Error cognitivo: Causado por los conceptos erróneos del escritor.

-Error fonético: Es el resultado de sustituir una secuencia de letras fonéticamente equivalentes.

Una solución propuesta por Gupta y Mathur (2012), es el análisis N-Grama. Este describe un método de detectar errores ortográficos en un grupo de texto. Se controlan usando una matriz n-dimensional donde se almacenan las frecuencias reales de n-gramas. Si encuentra uno inexistente o raro, la palabra se marca como falta, de lo contrario, no. Cabe señalar que un n-grama es un conjunto de caracteres consecutivos tomados de una cadena de longitud n que se establece previamente [12].

4.5 Uso de anáforas

La anáfora hace referencia a las expresiones en la representación lingüística. Estas normalmente ocurren por pronombres, los cuales reemplazan a una entidad previamente mencionada.

Por ejemplo, al decir: “Obtenga su tarjeta CuentaRUT, y deposite dinero en “ella”, el pronombre “ella” hace referencia a la tarjeta.

Por lo tanto, el problema se puede definir como: dada una representación lingüística R que carece de relaciones de dependencia referencial, y un conjunto A de antecedentes variables, da un antecedente en A para cada elemento anafórico en R. El conjunto A de los antecedentes disponibles modela el contexto del discurso en el que se produce el enunciado. El problema de decisión correspondiente es determinar si todos los elementos anafóricos en R pueden encontrar sus antecedentes en A[13].

4.6 Lenguaje coloquial

El lenguaje coloquial hace referencia a las conversaciones típicas diferentes a los textos formales, como los manuales, noticias y libros. Su uso implica abreviación de palabras, e infracciones a las reglas lingüísticas que se encuentran en el lenguaje escrito[14].

Por ejemplo, un usuario puede usar la expresión: “Al reponer mi tarjeta CuentaRUT, ¿me la entregarán altiro?”. La palabra “altiro” es propia del lenguaje coloquial chileno que significa “de inmediato”.

4.7 Anglicismo

La influencia del inglés en otros idiomas es un fenómeno habitual, y en el español no es la excepción. Palabras como *link*, *software*, entre otras, son palabras en inglés que se usan mucho en el idioma castellano, como la expresión: “¿Cuál es el link de la Banca en línea de BancoEstado?”. Como lo demostró Pratt [15], el inglés está presente en todos los niveles del sistema: ortográfico, léxico, semántico, morfológico y sintáctico. Como se sabe, la identificación del lenguaje escrito (LID) es la tarea de determinar el idioma en un documento electrónico. Se examinará cuatro enfoques de LID que están diseñados para tratar un texto de entrada en varios idiomas:

- Método basado en el análisis morfosintáctico combinado con búsqueda de léxico: Pfister y Romsdorfer (2003) describen los fenómenos de mezcla de lenguaje que se encuentran típicamente en los textos alemanes y derivan un método para analizar tales datos. Con diversos análisis se llegó a la conclusión de se requiere un análisis morfológico y sintáctico para procesar inclusiones foráneas para la síntesis del TTS (Text-to-Speech) [16].
- Diversos métodos para la identificación del lenguaje en textos multilingües: Marcadet, Fishcer y Waast-Richard (2005) realizaron un enfoque del LID diseñado específicamente para funcionar en el front-end de un sistema de síntesis TTS políglota. Su proyecto presenta experimentos con un aprendizaje basado en un diccionario basado en transformación de aprendizaje -en inglés, ***dictionary-based transformation-based learning*** (TBL)- y un enfoque del corpus basado en n-gram. Luego se muestran la combinación de ambos métodos.
- Hidden Markov Model Language Tagging: Farrugia (2005), propone la identificación del lenguaje basada en tokens para mensajes SMS en varios idiomas, mediante el etiquetado del lenguaje ***Hidden Markov Model*** (HMM) combinado con la búsqueda del diccionario y un modelo de lenguaje ***n-gram*** para tratar palabras desconocidas.
- La mayoría de los sistemas LID convencionales tienen éxito en el reconocimiento del lenguaje de porciones más grande de texto, pero no son adecuados para clasificar **tokens** individuales o subpartes de los mismos, chatbot muy limitados con las entradas.

4.8 Uso de chatbot en educación

Actualmente el uso de esta tecnología en el contexto de la educación es un tema que está tomando fuerza con el paso del tiempo, dado que en la actualidad el uso de tecnología de mensajería como WhatsApp, telegram entre otras. Dado esta realidad existe un alta de probabilidad de que los chatbot tengan un gran grado de aceptación en los estudiantes. Estudios realizados en el contexto de la docencia universitaria muestran que parte de los estudiantes tiene interacción con los chatbot de una manera regular, sin ocuparlo a deshoras, mientras que otro porcentaje de estudiantes los ocupa días antes de los exámenes. Lo que apunta que el uso de los chatbot depende la necesidad que el estudiante requiera [17].

5 Propuesta de solución

A continuación se presenta la propuesta de solución al problema encontrado, para la cual se dará a conocer el plan de trabajo tentativo, las herramientas de trabajo necesarias para

llevarlo a cabo, donde se utilizara Pandorabots el cual recordemos que trabaja con AIML, lenguaje de programación que será necesario para poder crear un flujo conversacional y Dialogflow que será necesario para poder trabajar con las entidades encontradas en la encuesta mixta y de esta manera poder crear un chatbot (Probot), más intuitivo con las entradas generadas por el usuarios.

5.1 Plan de trabajo tentativo:

En la primera etapa se planea obtener datos mediante una encuesta mixta (de respuesta cerrada y abierta), para elaborar una plantilla desde la que se obtendrán las entidades más mencionadas por los estudiantes y además con la encuesta se podrá generar el contexto conversacional, a partir de los datos recopilados en la encuesta aplicada. De manera posterior se empezará con el desarrollo de Probot y la generación de un prototipo para probar el funcionamiento en base a la plantilla creada y ver cómo es la aceptación de los alumnos al momento de utilizar Probot. Teniendo un prototipo ya elaborado, deberá pasar por una etapa de **Testing** y depuración para corrección de errores, la cual se planea de manera tentativa con los alumnos de programación el segundo semestre del presente año. Con las correcciones ya hechas se pasará a la fase final del desarrollo de Probot, obteniendo como resultado la versión final de Probot.

5.2 Encuesta mixta

La encuesta mixta o semiestructurada es aquella en la que, el encuestador despliega una estrategia mixta, alternando preguntas estructuradas con preguntas espontáneas. Esta es la forma más completa ya que, mientras que la parte preparada permite comparar entre los diferentes examinados y la parte libre permite profundizar en las características específicas de cada examinado. Por ello, permite una mayor libertad y flexibilidad en la obtención de información.

5.3 Herramientas de desarrollo

A continuación, se presentan las herramientas que fueron seleccionadas para llevar a cabo el desarrollo de Probot.

5.3.1 Dialogflow

Con la encuesta mixta realizada, en la cual se obtendrán las entidades más mencionadas por los alumnos. Se ha seleccionado trabajar con Dialogflow dado que la manera de trabajar de esta plataforma, permite crear un bot más permisivo en la manera de realizarle una pregunta, ya que hay diferentes maneras de realizar la misma pregunta y con esta plataforma es posible realizar aquella tarea de manera fácil y menos engorrosa.

5.3.2 Pandorabots

Con el fin de generar un contexto conversacional ideal, donde se pregunte principalmente casos puntuales, como por ejemplo definiciones conceptuales puntuales de contenidos, se ha seleccionado ocupar Pandorabots que trabaja con AIML.

De esta manera ante una entrada del usuario que coincida con el flujo conversacional creado se podrá responder de una manera óptima, ya que AIML podremos enfrentar a un caso ideal donde, pregunte lo que uno quiera dicho de una manera coloquial.

5.4 Marcha blanca de Probot

En una etapa inicial se va a generar un prototipo de Probot, el que se pretende probar con los alumnos del curso de fundamentos de programación, y con ello se espera obtener un feedback y en base al flujo conversacional creado en el caso de pandorabots y por otro lado ver si las entidades que utilizamos en Dialogflow están siendo utilizadas de manera correcta, en las entradas generadas por los alumnos. Recordando que las entidades en el caso de Dialogflow se ocupan poder saber en qué contexto estamos trabajando y poder tomar las acciones y dar una respuesta acorde a la entrada.

Ya habiendo ejecutado esta etapa se pasará a la depuración de los errores encontrados y a su corrección para ya dar paso a una segunda versión obteniendo un flujo conversacional mejorado y las entidades ya depuradas.

6 Diseño

A continuación, se dan a conocer las herramientas, framework y base de datos requeridas durante el desarrollo de probot.

6.1 Interfaz

Para el diseño de la interfaz de Probot fue seleccionado Angular, como es una plataforma enfocado en el desarrollo de frontend y en la parte estética fueron requeridos bootstrap y fontawesome.

6.1.1 Angular

Es una plataforma que facilita la creación de aplicaciones con la web, desarrolladas en TypeScript de código abierto, es mantenida por Google. Angular se utiliza para crear y mantener aplicaciones web de una sola página, combina plantillas declarativas, inyección de dependencia. Permite a los desarrolladores crear aplicaciones que viven en la web, el móvil o el escritorio [18].

6.1.2 Bootstrap

Es un framework originalmente fue creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya característica es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de un ordenador/pc, Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como Responsive Design o Diseño Adaptativo [19].

6.1.3 Font Awesome

Es un framework de iconos vectoriales y estilos css. Este es utilizado para sustituir imágenes de iconos comunes por gráficos vectoriales convertidos en fuentes . Además permite reducir significativamente el uso de imágenes para iconos (lo cual se traduce en mejorar la velocidad de nuestra web) [20].

6.2 Base de datos

Fue necesario utilizar base datos para llevar un registro del feedback de Probot dado que es un prototipo. Con este feedback generado se puede ver apreciación de los usuarios(estudiantes) y si el procesamiento del texto se está llevando de manera adecuada, para realizar el registro se está utilizando firebase, la razón principal que se seleccionó firebase por su compatibilidad con angular y su documentación disponible en la web para poder realizar la conexión angular-firebase.

6.2.1 Firebase

Es una base de datos desarrollada por Google, la cual es una base no sql y no relacional, la cual tiene diversas funciones disponibles en la web entre las cuales podemos destacar Firebase Cloud Messaging, Firebase Auth, Realtime Database, Firebase Storage entre otras. Para esto proyecto se está ocupando el servicio Realtime Database donde la funcionalidad es que proporciona una base de datos en tiempo real y back-end. Este servicio proporciona a los desarrolladores de aplicaciones una API que permite que la información de las aplicaciones sea sincronizada y almacenada en la nube de Firebase [21].

6.3 Pantallazos Probot

En este ítem se muestran unos pantallazos de como fue el producto final aplicando todas las herramientas mencionadas, los cuales se pueden encontrar en el ítem de anexos.

7 Desarrollo del software

7.1 Obtención de base de conocimientos

Para la obtención de la base de conocimientos fue necesario realizar la encuesta mixta, donde se pudo tener una perspectiva del contenido que son más difícil de comprender por los estudiantes y en base a estos resultados se pudo crear el flujo conversacional (administrado por pandorabots) y los modelos de intents que posee el agente de dialogflow.

Un punto destacable de la obtención de la base de conocimientos, es que los resultados obtenidos de la encuesta aplicada a los estudiantes, no estaban lejanos al pronóstico que se tenía en un principio. Como por el ejemplo el contenido que más era difícil de comprender por los estudiantes era la materia asociada a punteros.

7.2 Implementación de objetivo general

A continuación, se presenta el objetivo principal del Probot y como se llevó a cabo tal propósito.

1. Desarrollar prototipo de chatbot(Probot), utilizando frameworks y/o librerías existentes, que soporte consultas básicas de estudiantes de cursos de programación de primer año para servir como herramienta de apoyo en el curso Fundamentos de Programación.

Para llevar a cabo este objetivo se investigó qué framework iba a ser el encargado de integrar las herramientas de desarrollo de chatbot donde fue seleccionado Angular el que integra pandorabots y dialogflow que son las encargadas de procesar el texto ingresado por el usuario y la base de datos que va a almacenar el feedback ingresado por los usuarios es firebase.

El tipo de consulta que va a soportar Probot son entradas básicas, entiendo que son preguntas conceptuales e informaciones específicas que se requieran.

7.3 Implementación de objetivos específicos

A continuación, se presenta una lista numerada de los objetivos específicos planteados en el avance del proyecto y una lista numerada correspondiente al objetivo numerado de cómo se pudo realizar.

Lista de objetivos específicos

Objetivo específico	¿Cómo se realizó?
Identificar principales dudas y/o inquietudes de los estudiantes de primer año sobre	El primer objetivo se llevó a cabo mediante la encuesta mixta para poder obtener datos

programación en el lenguaje C.	significativos de los estudiantes y tener un panorama de qué contenidos se debía poner énfasis, para agregarlos a la base de conocimiento en probot.
Determinar su aplicabilidad de estas dudas y/o inquietudes como parte del Chatbot.	En base a los resultados obtenidos, hubo que diferenciar en grado importancia que datos eran más significativos que otros.
Determinar las preguntas que serán respondidas con el modelo de Entidades, y las que serán respondidas por el modelo de Flujo Conversacional.	Con los datos obtenidos se llegó a la decisión que las preguntas relacionadas con la materia del curso como definiciones conceptuales, ejemplos entre otras. Serán respondidas por dialogflow y las informaciones del curso serán respondidas por pandorabots
Diseñar las entidades y flujos conversacionales según corresponda a cada pregunta.	Teniendo los datos ordenados, se procedió a generar las estructuras de procesamiento de texto, donde la herramienta pandorabots va a poseer el conocimiento, con lo que respecta al ramo de fundamentos de programación, dado que son informaciones puntuales y pandorabots trabaja con el sistema pregunta y respuesta satisface esta necesidad y el conocimiento de definiciones conceptuales lo almacena a dialogflow dado que su procesamiento de texto es más complejo al poseer machine-learning (Aprendizaje de máquina) y muchas maneras de preguntar alguna definición y/o concepto.
Elaborar Prototipo de chatbot que resuelva principales dudas respecto a programación.	Para la elaboración de Probot se ocupó Angular versión 5.1.2 para mezclar las 2 herramientas de desarrollo de bots de tal manera que al momento de ingresarse una pregunta se realiza un recorrido por el conocimiento de pandorabots y en caso de no tener respuesta a la pregunta ingresada se llama a dialogflow para dar una respuesta ante la entrada del usuario. El motivo de

	porque se seleccionó angular y no otro framework existente, fue como angular está enfocado en el desarrollo de frontend y el backend vendría siendo pandorabots y dialogflow, como está desarrollado por Google tiene una amplia documentación de cómo integrarse con angular que también está desarrollado por Google.
Evaluar de manera experimental la apreciación de los estudiantes en cuanto a la utilidad del Chatbot, mediante su aplicación en actividades controladas.(Mediante registro en base de datos).	Este objetivo no está cumplido en su totalidad dado que no llegó a ser probado en estudiante, lo que está terminado es el registro de feedback, el cual se desarrolló en firebase, fue ocupado firebase como también está desarrollado por Google y su integración con angular es bastante amigable por la documentación disponible en la web.

Tabla 7.3.1 Implementación de objetivos específicos.

8 Alcances y limitaciones

8.1 Alcances

El alcance de probot es llegar a procesar el texto de preguntas básicas realizadas por los estudiantes, específicamente informaciones puntuales, definiciones conceptuales de lenguaje C, informaciones del ramo de fundamentos de programación, de tal manera que al momento

de ingresar alguna de estas preguntas se entrega la respuesta correspondiente a la entrada ingresada.

8.2 Limitaciones

La principal limitación que tiene Probot, es que las respuestas solo pueden ser string básicos, no puede dar como output imágenes, videos tutoriales y string tipo url.

La base de conocimiento aún se encuentra en construcción por lo que el dominio que probot posee es acotado en esta etapa de su desarrollo.

8.3 Trabajo a futuro

Para trabajo a futuro se buscará mejorar el procesamiento del texto de tal manera que las preguntas ingresadas por el usuario sean más variadas y no se limiten a casos puntuales de esta manera las respuestas entregadas por probot puedan ser más variadas como link, imágenes, videos tutoriales y poder seguir un flujo conversacional de una manera más natural sin tanta restricción y se entrega la respuesta deseada por el usuario de una manera más rápida.

Además, poder agregar un registro de las conversaciones con los usuarios para poder tener cada vez una conversación más fluida y teniendo ciertos patrones de respuesta ya almacenados. Un tema que se vio dentro de la investigación fue el tema de la Ontología el cual es un concepto que sirve para darle un grado de complejidad más alto al procesamiento del texto y poder llegar a una respuesta más acertada.

9 Conclusiones

En conclusión, el objetivo general y específicos del proyecto fueron cumplidos dentro de lo esperado. Donde se pudo conseguir integrar las 2 herramientas para poder generar un chatbot que tiene un procesamiento de texto más complejo e a la vez más completo, dado que se está mezclando 2 herramientas para el procesamiento de texto dialogflow y pandorabots, se prefirió ocupar estas herramientas porque trabajan con 2 formas distintas de procesamiento de texto, pero se complementan de buena manera. Por otro lado el trabajo se vio limitado en el tipo de entradas que se pudo lograr a procesar dado que solo soporta entradas de tipo básicas, entendiendo como pregunta básica preguntas conceptuales e informaciones generales del

curso de fundamentos de programación, pero con el panorama actual en el que quedó Probot, se puede concluir que el alcance de este proyecto es alto, porque el hecho de mejorar el procesamiento del texto, agregar más conocimiento al bot, tiene muchos conceptos que pueden investigados e integrados a probot para cada vez, ir obteniendo un bot cada vez más complejo y funcional el cual pueda ser un futuro un herramienta de apoyo para los estudiantes de la PUCV en general, teniendo un dominio de conocimiento amplio.

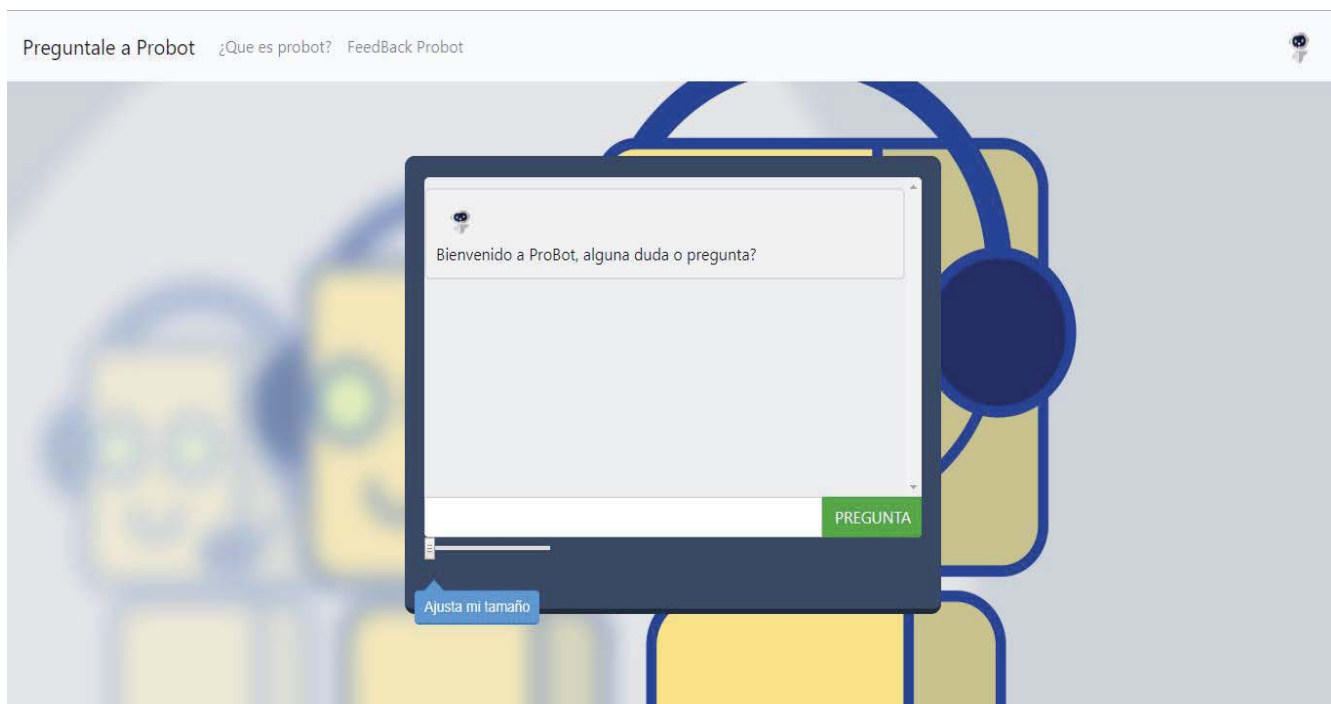
Además, en cada mejora que se vaya realizando a probot se podrá ir mejorando la usabilidad de este haciendo una herramienta multiplataforma, responsive entre otras y poder brindar una mejor experiencia usuario, pudiendo dar la respuesta de una manera rápida, exacta y concisa.

10 Bibliografía:

- [1] B. AbuShawar and E. Atwell, “ALICE Chatbot: Trials and Outputs,” *Comp. y Sist.*, vol. 19, no. 4, pp. 625–632, 2015.
- [2] “<https://dialogflow.com/docs/>.”
- [3] “<https://dialogflow.com/docs/intro/>.”
- [4] “Introduction.” [Online]. Available: <https://get.converse.ai/docs>. [Accessed: 16-Jan-2018].
- [5] “AIML Fundamentals · Pandorabots Documentation.” [Online]. Available: <https://pandorabots.com/docs/aiml/aiml-fundamentals.html>. [Accessed: 01-Feb-2018].
- [6] “About Pandorabots.” [Online]. Available: <https://www.pandorabots.com/docs/>. [Accessed: 16-Jan-2018].

- [7] P. E. Alcaraz and M. A. Martínez, *Diccionario de Lingüística Moderna*, vol. 1. Ariel, Mayo de 1997.
- [8] M. Mozgovoy, “Dependency-Based Rules for Grammar Checking with LanguageTool,” in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 2011, pp. 209–212.
- [9] A. Ratnaparkhi, “A maximum entropy model for part-of-speech tagging,” in *Proceedings of the conference on empirical methods in natural language processing*, 1997, pp. 133–142.
- [10] J. C. Reynar and A. Ratnaparkhi, “A maximum entropy approach to identifying sentence boundaries,” in *Proceedings of the fifth conference on Applied natural language processing -*, 1997.
- [11] D. Hladek, J. Stas, and J. Juhar, “Unsupervised Spelling Correction for Slovak,” *Advances in Electrical and Electronic Engineering*, vol. 11, no. 5, 2013.
- [12] N. Gupta and P. Marthur, “Spell Checking Techniques in NLP: A Survey,” in *IJARCSSE*, December, 12 2012, pp. 217–221.
- [13] E. S. Ristad, “The Anaphora Problem,” *Inform. and Comput.*, vol. 105, no. 1, pp. 105–131, 1993.
- [14] F. Popowich, D. Turcato, O. Laurens, and P. McFetridge, “A Lexicalist Approach to the Translation of Colloquial Text,” *School of Computing Science Simon Fraser University*.
- [15] C. Pratt, *El anglicismo en el español peninsular contemporáneo*. 1980.
- [16] H. Romsdorfer and B. Pfister, “Text analysis and language identification for polyglot text-to-speech synthesis,” *Speech Commun.*, vol. 49, no. 9, pp. 697–724, 2007.
- [17] “Uso de Chatbots en la Docencia Universitaria.”
- [18] “Angular-document.” [Online]. Available: <https://angular.io/docs>.
- [19] “Bootstrap_doc.” [Online]. Available: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>.
- [20] “<https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>.” [Online]. Available: <https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>.
- [21] “firebase-document.” [Online]. Available: <https://firebase.google.com/docs/?authuser=0>.

Anexos



Pantallazo pantalla inicial Probot.



Pantallazo pantalla introductoria a que es Probot.

Preguntale a Probot ¿Que es probot? FeedBack Probot

Agregar Comentario

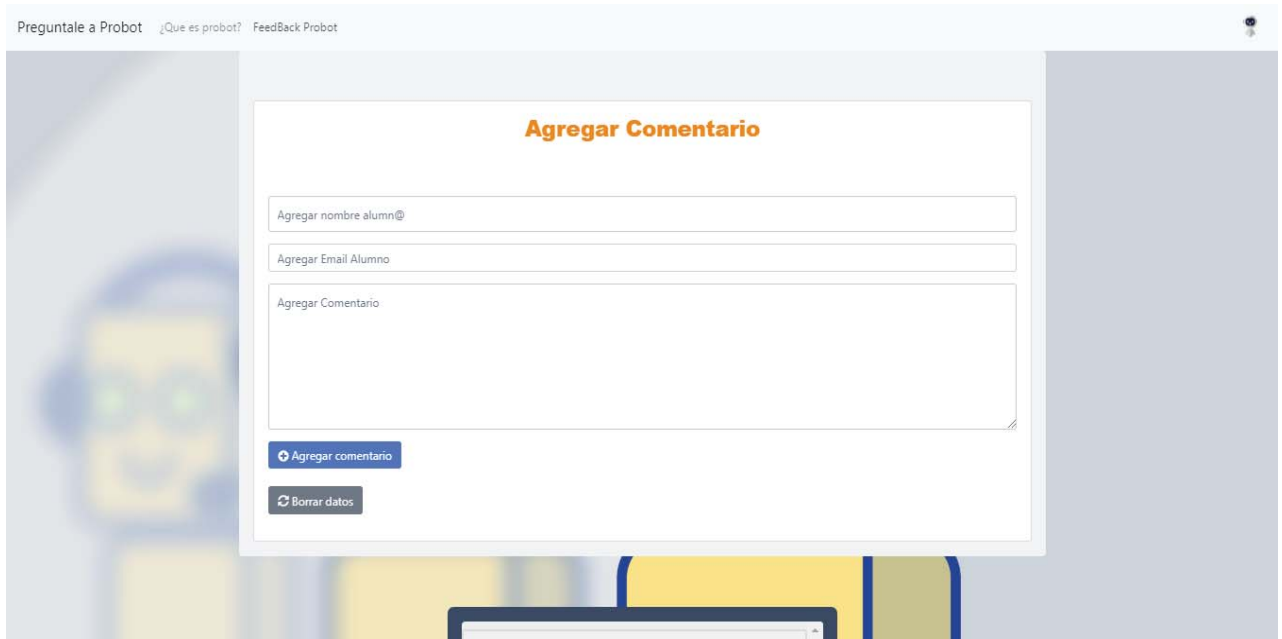
Agregar nombre alumn@

Agregar Email Alumno

Agregar Comentario

+ Agregar comentario

🔄 Borrar datos



Pantallazo pantalla feedback Probot.